CHAPTER 4

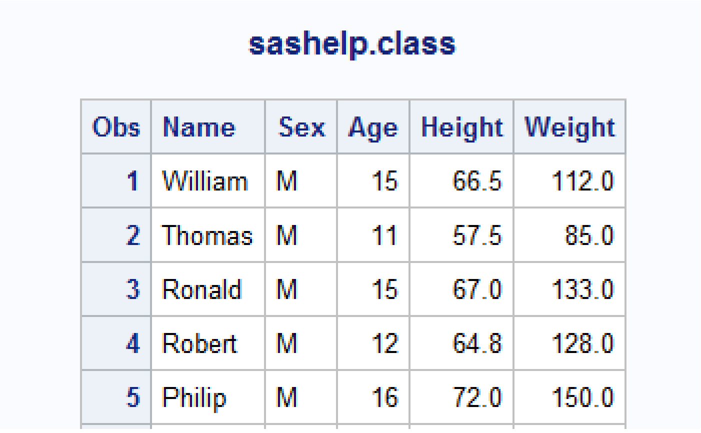
Customized Pointlabels

Purpose: This chapter demonstrates how to create customized pointlabels, using annotate.

GPlot Labels

We used the SYMBOL statement’s POINTLABEL option to add text labels to certain plot markers in the “MPG Analysis” in Chapter 1 of *SAS/Graph: The Basics*. But what if you want to customize your pointlabels? You can customize them a bit with the POINTLABEL option (font, color, size, justification), but if you really want to take full control, you’ll need to use annotate.

For the examples in this chapter, we’ll use the sashelp.class dataset (which ships with SAS). Here’s what the first few rows of the data look like:



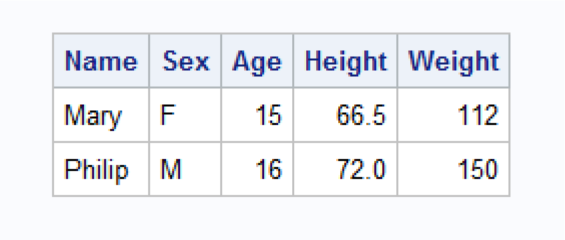
sashelp.class

Obs	Name	Sex	Age	Height	Weight
1	William	M	15	66.5	112.0
2	Thomas	M	11	57.5	85.0
3	Ronald	M	15	67.0	133.0
4	Robert	M	12	64.8	128.0
5	Philip	M	16	72.0	150.0

Let’s start with something simple – annotating a text label directly above the plot markers, for the tallest male and female in the class. I like using Proc SQL to determine things like min & max, and create a dataset with the results.

```
proc sql;
create table my_summary as
select unique *
from sashelp.class
group by sex
having height=max(height);
quit; run;
```

Here is what the resulting data (my_summary) looks like, containing just the observations for the tallest male and female:

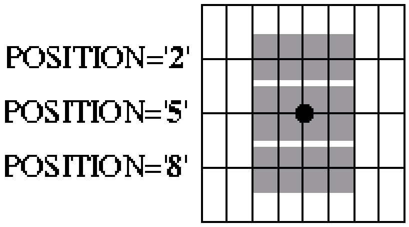


Name	Sex	Age	Height	Weight
Mary	F	15	66.5	112
Philip	M	16	72.0	150

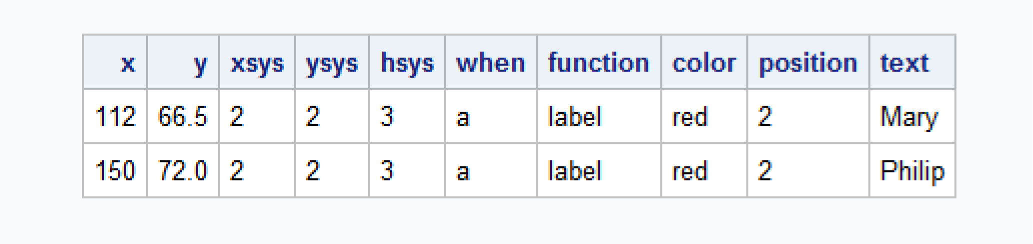
Now let’s convert that dataset (my_summary) into an annotate dataset that we’ll call my_anno. The xsys and ysys values tell the annotation to use the same coordinate system as the data, and we set x to the weight and y to the height. We’re using the ‘label’ annotate function, and the position=’2’ places the label directly above the x/y coordinate.

```
data my_anno; set my_summary;
length function $8 text $20;
xsys='2'; ysys='2'; hsys='3'; when='a';
x=weight; y=height;
text=trim(left(name));
function='label'; color='red'; position='2';
run;
```

There are several possible values for annotate position, that control the placement and justification of the text, in relation to the x/y coordinate. As shown in the illustration below, position=’2’ is above the x/y coordinate and centered left/right.



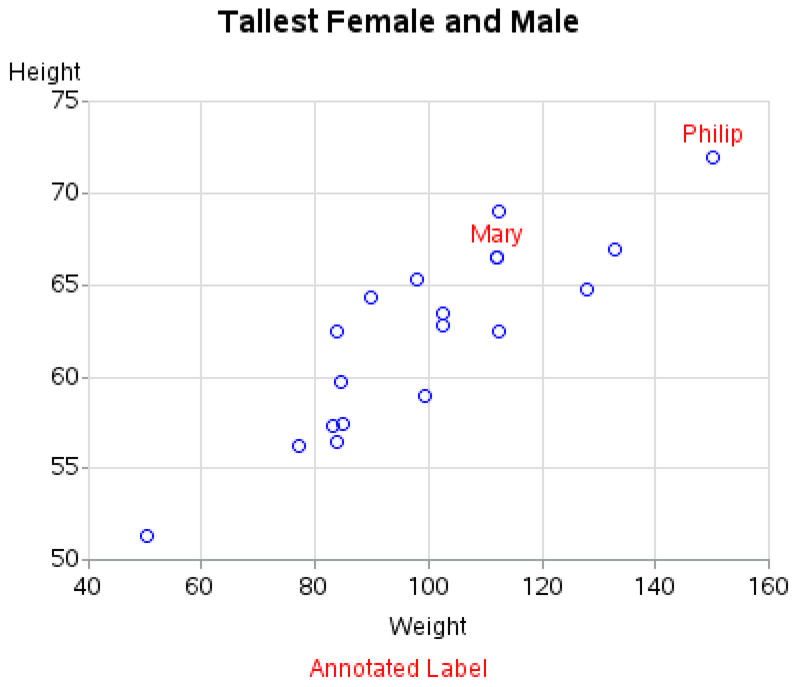
Here is the my_anno dataset, created by the above code:



x	y	xsys	ysys	hsys	when	function	color	position	text
112	66.5	2	2	3	a	label	red	2	Mary
150	72.0	2	2	3	a	label	red	2	Philip

We can now create a Proc GPlot scatter, specifying the annotate dataset, and get the desired labels above the markers:

```
title1 ls=1.5 "Tallest Female and Male";
symbol1 value=circle height=3 interpol=none color=blue;
axis1 order=(50 to 75 by 5) minor=none offset=(0,0);
axis2 order=(40 to 160 by 20) minor=none offset=(0,0);

proc gplot data=sashelp.class anno=my_anno;
plot height*weight /
 vaxis=axis1 haxis=axis2 noframe
 autovref cvref=graydd
 autohref chref=graydd;
run;
```



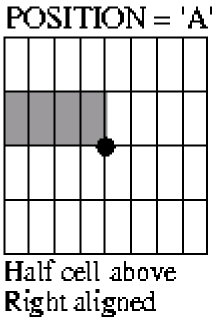
✓-

So far, the results are not much different from what we could get with the POINTLABEL option, but now that we are using annotate, we have full control. For example, it's difficult to control the exact position of the pointlabel text labels because of the automatic collision avoidance - but with annotated labels, you can specify exactly where you want the labels.

In the previous graph, for example, it might be difficult for a user to know if the 'Mary' label goes with the marker below it, or the marker above it. If we move Mary's label to the left of the desired marker, then it will be easier to tell which marker the label goes with. We can easily do that with annotate, specifying that position for the 'Mary' label in the code as follows:

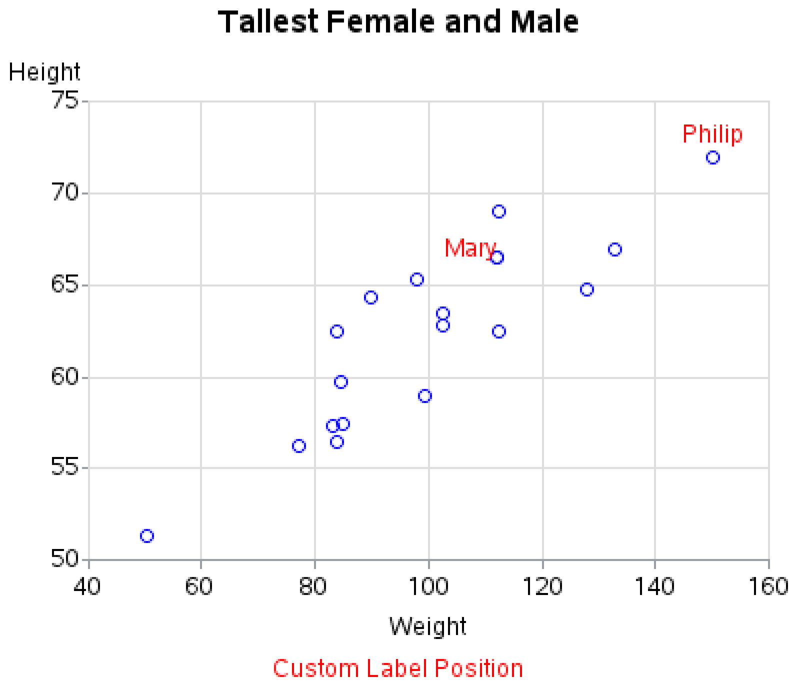
```
data my_anno; set my_summary;
length function $8 text $20;
xsys='2'; ysys='2'; hsys='3'; when='a';
x=weight; y=height;
text=trim(left(name));
function='label'; color='red';
position='2';
if name='Mary' then position='a';
run;
```

Note that position='a' is to the top left of the x/y coordinate, and right-justified:



```
title1 ls=1.5 "Tallest Female and Male";
symbol1 value=circle height=3 interpol=none color=blue;
axis1 order=(50 to 75 by 5) minor=none offset=(0,0);
axis2 order=(40 to 160 by 20) minor=none offset=(0,0);

proc gplot data=sashelp.class anno=my_anno;
plot height*weight /
 vaxis=axis1 haxis=axis2 noframe
 autovref cvref=graydd
 autohref chref=graydd;
run;
```

The ‘Mary’ label is now to the left of the marker, but it still doesn’t look great. The text overlaps the marker, so we need to do a bit of fine-tuning …



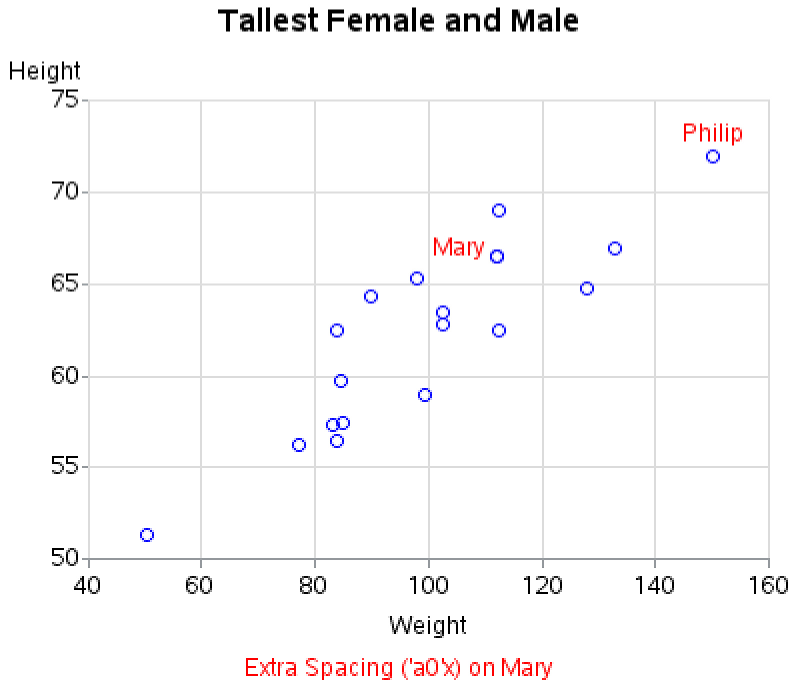
✓-

The annotate positions allow for general placement and justification of text labels around your x/y point, but for fine-adjustments I often pad blank spaces onto the text labels. But you can't use a regular ' ' blank, because those get trimmed by annotate. Rather than using a regular blank (which is hex code '20'x) I pad the text with a no-break space (which is hex code 'a0'x). In the code below, I pad two such characters to the right-side of 'Mary'.

```
data my_anno; set my_summary;
length function $8 text $20;
xsys='2'; ysys='2'; hsys='3'; when='a';
x=weight; y=height;
text=trim(left(name));
function='label'; color='red';
position='2';
if name='Mary' then do;
 position='a';
 text=trim(left(text))||'a0a0'x;
 end;
run;
```

And now the 'Mary' text label does not overlap the marker:

```
title1 ls=1.5 "Tallest Female and Male";
symbol1 value=circle height=3 interpol=none color=blue;
axis1 order=(50 to 75 by 5) minor=none offset=(0,0);
axis2 order=(40 to 160 by 20) minor=none offset=(0,0);

proc gplot data=sashelp.class anno=my_anno;
plot height*weight /
 vaxis=axis1 haxis=axis2 noframe
 autovref cvref=graydd
 autohref chref=graydd;
 run;
```



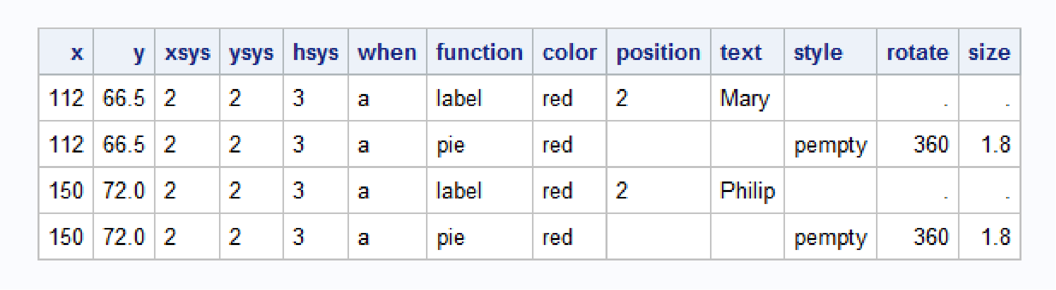
✓

The ability to annotate the labels exactly where you want them gives you a slight advantage over the standard pointlabels … but now let's annotate some 'graphics' along with the text, to take it to the next level!

In this next example, we'll use the annotate 'pie' function to draw a circle around the marker, to make it more evident exactly which marker is of interest. We'll make the pie 'pempty' (as opposed to 'psolid') so that it is just an outline/circle around the marker.

```
data my_anno; set my_summary;
length function $8 text $20;
xsys='2'; ysys='2'; hsys='3'; when='a';
x=weight; y=height;
text=trim(left(name));
function='label'; color='red'; position='2';
output;
function='pie'; rotate=360; size=1.8;
 position=''; text=''; style='pempty';
output;
run;
```

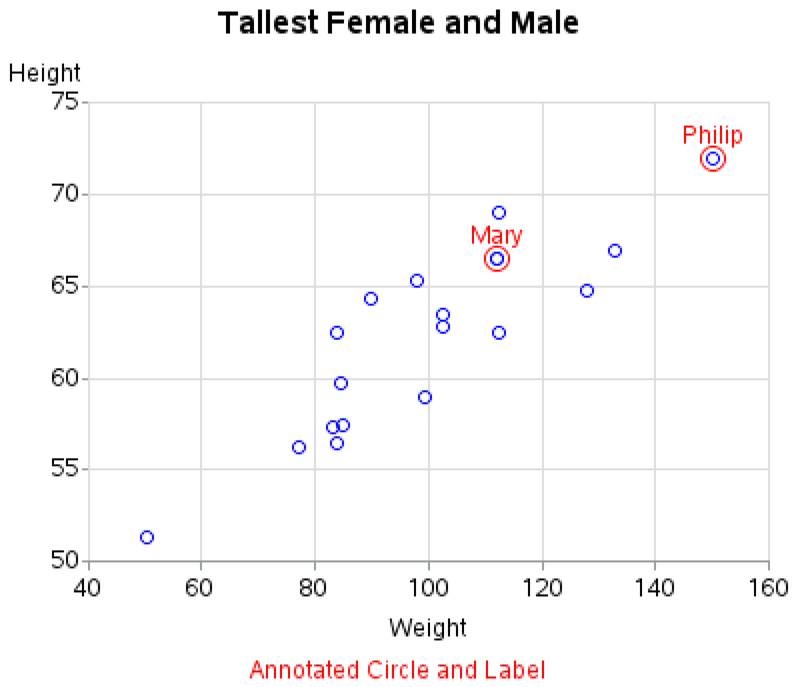
This is what the annotate data set looks like now – notice that certain rows of the data draw the text, and certain rows draw the circle:



x	y	xsys	ysys	hsys	when	function	color	position	text	style	rotate	size
112	66.5	2	2	3	a	label	red	2	Mary		.	.
112	66.5	2	2	3	a	pie	red			pempty	360	1.8
150	72.0	2	2	3	a	label	red	2	Philip		.	.
150	72.0	2	2	3	a	pie	red			pempty	360	1.8

And when we plot the data, we get the following scatter plot with the markers of interest both labeled and circled. Combining the text labels and the graphical circles makes a much stronger visual impact that just the text labels alone.

```
title1 ls=1.5 "Tallest Female and Male";
symbol1 value=circle height=3 interpol=none color=blue;
axis1 order=(50 to 75 by 5) minor=none offset=(0,0);
axis2 order=(40 to 160 by 20) minor=none offset=(0,0);

proc gplot data=sashelp.class anno=my_anno;
plot height*weight /
 vaxis=axis1 haxis=axis2 noframe
 autovref cvref=graydd
 autohref chref=graydd;
run;
```



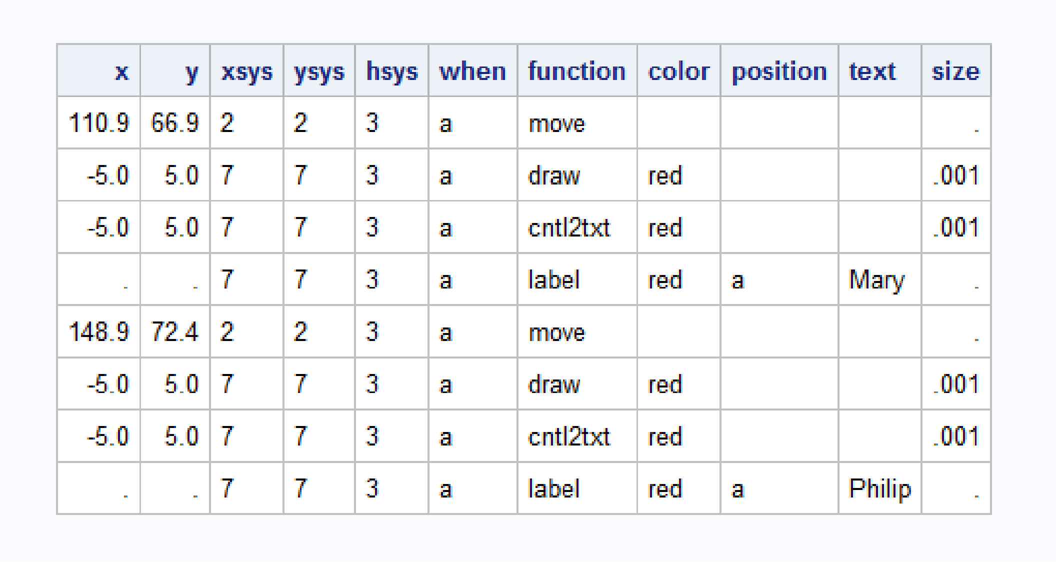
✓+

Instead of circling the markers of interest, what if we want to draw a line from the marker to the label? We can do that with annotate by first using the ‘move’ command to move to the center of the marker (actually, in this case I add a slight offset so I’m moving just to the top/left side of the marker), and then using the annotate ‘draw’ command to draw a line up and to the left of the marker. I then use the ‘cntl2txt’ function so I can start writing the text label at the end of the line.

```
data my_anno; set my_summary;
length function $8 text $20;
xsys='2'; ysys='2'; hsys='3'; when='a';
x=weight-1.1; y=height+.4;
function='move'; output;
xsys='7'; ysys='7';
x=-5; y=5;
function='draw'; color='red'; size=.001; output;
function='cntl2txt'; output;
x=.; y=.;
```

```
function='label'; color='red'; size=.; position='a';
text=trim(left(name));
output;
run;
```

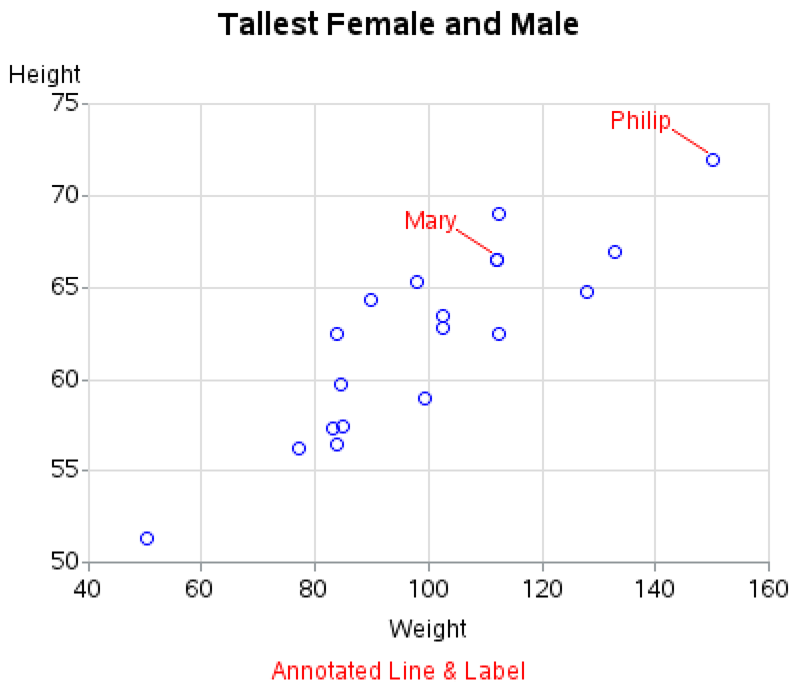
The above code produces the following annotate data set:



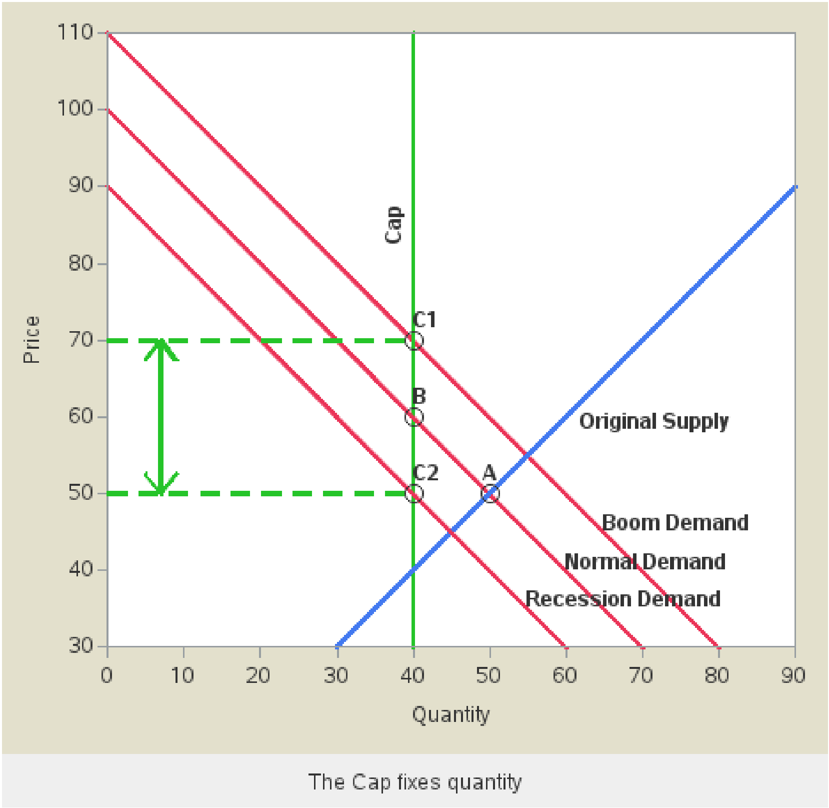
x	y	xsys	ysys	hsys	when	function	color	position	text	size
110.9	66.9	2	2	3	a	move				.
-5.0	5.0	7	7	3	a	draw	red			.001
-5.0	5.0	7	7	3	a	cntl2txt	red			.001
.	.	7	7	3	a	label	red	a	Mary	.
148.9	72.4	2	2	3	a	move				.
-5.0	5.0	7	7	3	a	draw	red			.001
-5.0	5.0	7	7	3	a	cntl2txt	red			.001
.	.	7	7	3	a	label	red	a	Philip	.

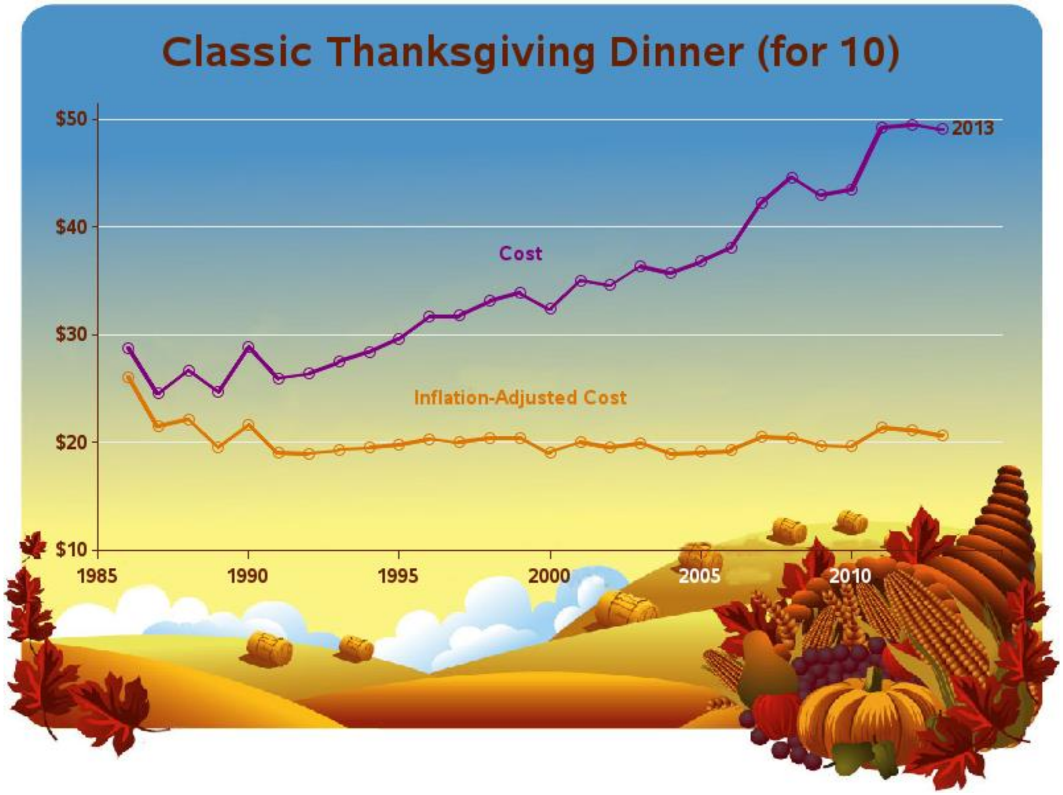
And now when you plot the data, you get these very nice labels, with lines showing which markers they go with.

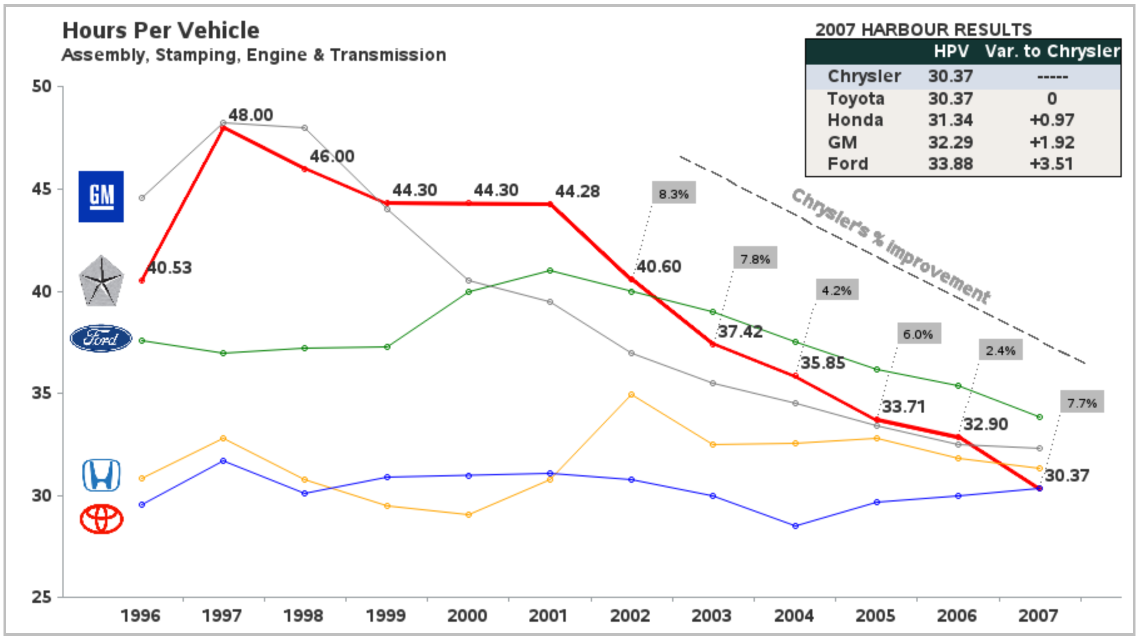
```
title1 ls=1.5 "Tallest Female and Male";
symbol1 value=circle height=3 interpol=none color=blue;
axis1 order=(50 to 75 by 5) minor=none offset=(0,0);
axis2 order=(40 to 160 by 20) minor=none offset=(0,0);

proc gplot data=sashelp.class anno=my_anno;
plot height*weight /
 vaxis=axis1 haxis=axis2 noframe
 autovref cvref=graydd
 autohref chref=graydd;
run;
```



Below are some example of SAS graphs with custom annotated pointlabels.

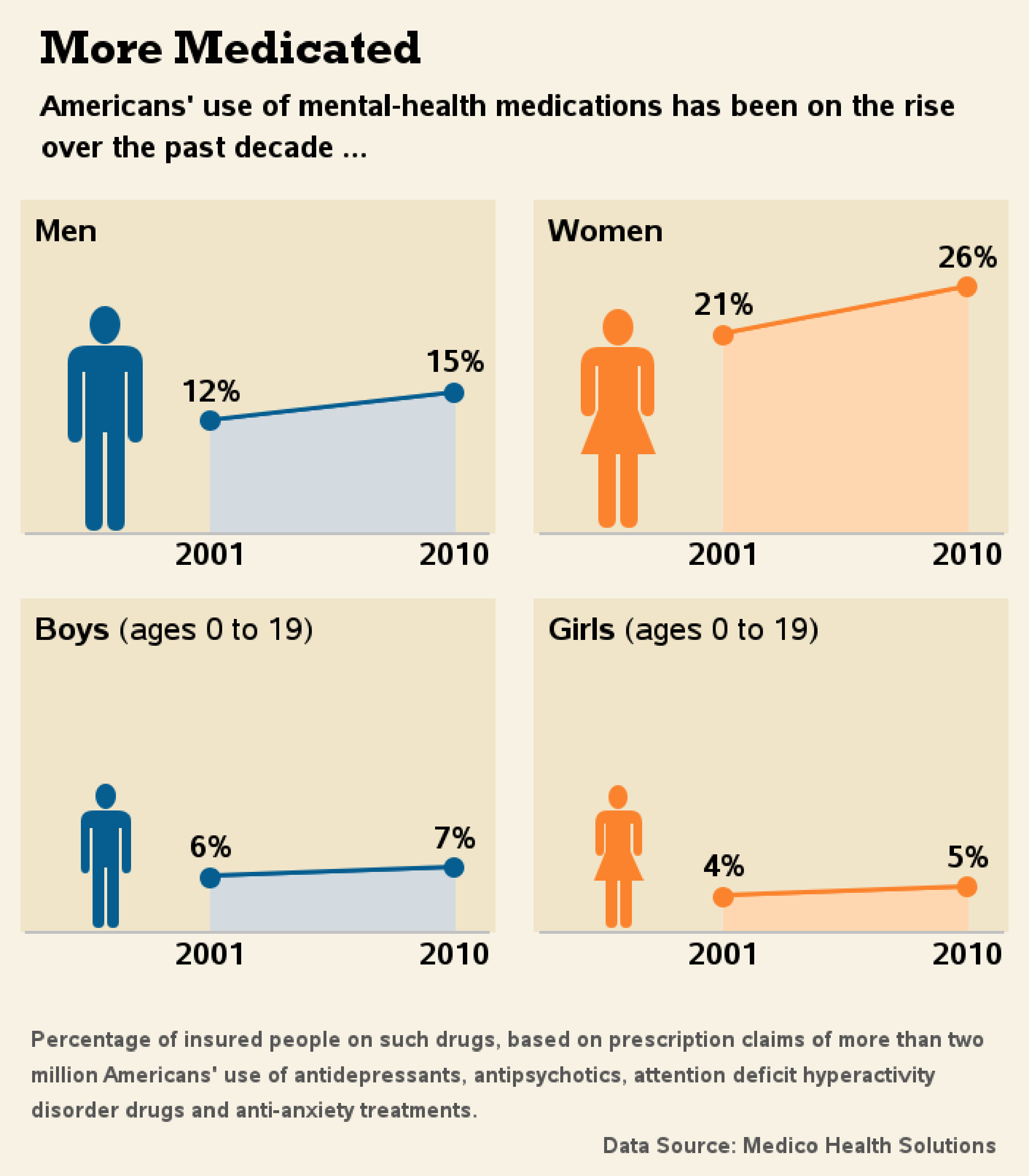






2007 HARBOUR RESULTS

	HPV	Var. to Chrysler
Chrysler	30.37	-----
Toyota	30.37	0
Honda	31.34	+0.97
GM	32.29	+1.92
Ford	33.88	+3.51



More Medicated

Americans' use of mental-health medications has been on the rise over the past decade ...

Percentage of insured people on such drugs, based on prescription claims of more than two million Americans' use of antidepressants, antipsychotics, attention deficit hyperactivity disorder drugs and anti-anxiety treatments.

Data Source: Medico Health Solutions